CLAIMS

What is claimed is:

1. A system having secure system firmware, comprising:
a central processing unit (CPU);
a dynamic random access memory (DRAM) coupled to the CPU that comprises
a shadow random access memory (RAM) including one or more registers whose
5   attributes are separately configurable; and
system firmware that when the system is reset, initializes the DRAM and the
shadow RAM, copies itself into the shadow RAM, sets LOCK bits associated with the
registers of the shadow RAM, boots a computer operating system, monitors attempted
writes to locked registers of the shadow RAM, and if a write operation to a locked
10   register is detected, generates an interrupt that indicates an attempt to tamper with the
system firmware.

2. The system recited in Claim 1 wherein the interrupt that is generated is
selected from a group consisting of a system management interrupt (SMI), a non-
maskable interrupt (NMI) and a general-purpose interrupt.

3. The system recited in Claim 1 wherein the system firmware enables
generation of the interrupt before initiating operating system code and after all
modifications to the shadow RAM are complete.

4. The system recited in Claim 1 wherein the system firmware begins execution
when the interrupt is generated and performs a desired behavior.

5. The system recited in Claim 4 wherein the desired behavior includes an
security alert, remote administrator signaling, logging of an event, or ignoring of the
event and resuming operation.

6. The system recited in Claim 1 wherein the system firmware is selectively
configured to programmatically enable and disable write access to a selected shadow
RAM register, programmatically enable and disable read access to a selected shadow
RAM register, and programmatically enable and disable cacheability of a shadow RAM
5   register.

7. A method for use with a computer system having a central processing unit (CPU), a dynamic random access memory (DRAM) coupled to the CPU that comprises a shadow random access memory (RAM) including one or more registers whose attributes are separately configurable, and system firmware that runs on the CPU, the method comprising the steps of:

    initializing the DRAM and the shadow RAM;

    copying itself into the shadow RAM;

    setting LOCK bits associated with the registers of the shadow RAM;

    booting a computer operating system;

    monitors attempted writes to locked registers of the shadow RAM; and

    if a write operation to a locked register is detected, generating an interrupt that indicates an attempt to tamper with the system firmware.

8. The method recited in Claim 7 wherein the interrupt that is generated is selected from a group consisting of a system management interrupt (SMI), a non-maskable interrupt (NMI) and a general-purpose interrupt.

9. The method recited in Claim 7 wherein the system firmware generates 47 the interrupt before initiating operating system code and after all modifications to the shadow RAM are complete.

10. The method recited in Claim 7 wherein the system firmware begins execution when the interrupt is generated and performs a desired behavior.

11. The method recited in Claim 10 wherein the desired behavior includes an security alert, remote administrator signaling, logging of an event, or ignoring of the event and resuming operation.

12. The method 40 recited in Claim 7 wherein the system firmware is selectively configured to programmatically enable and disable write access to a selected shadow RAM register, programmatically enable and disable read access to a selected shadow RAM register, and programmatically enable and disable cacheability of a shadow RAM register.

13. Software for use with a computer system having a central processing unit (CPU), a dynamic random access memory (DRAM) coupled to the CPU that comprises a shadow random access memory (RAM) including one or more registers whose

attributes are separately configurable, and system firmware that runs on the CPU, that
5   comprises:

a code segment that initializes the DRAM and the shadow RAM;

a code segment that copies itself into the shadow RAM;

a code segment that sets LOCK bits associated with the registers of the shadow RAM;

10   a code segment that boots a computer operating system;

a code segment that monitors attempted writes to locked registers of the shadow RAM; and

a code segment that, if a write operation to a locked register is detected, generates an interrupt that indicates an attempt to tamper with the system firmware.

14. The software recited in Claim 13 wherein the interrupt that is generated is selected from a group consisting of a system management interrupt (SMI), a non maskable interrupt (NMI) and a general-purpose interrupt.

15. The software recited in Claim 13 wherein the interrupt generating code segment generates the interrupt before initiating operating system code and after all modifications to the shadow RAM are complete.

16. The software recited in Claim 13 further comprising a code segment that begins execution when the interrupt is generated and performs a desired behavior.

17. The software recited in Claim 16 wherein the desired behavior includes an security alert, remote administrator signaling, logging of an event, or ignoring of the event and resuming operation.

18. The software recited in Claim 13 further comprising a code segment that programmatically enable and disable write access to a selected shadow RAM register.

19. The software recited in Claim 13 further comprising a code segment that programmatically enables and disables read access to a selected shadow RAM register.

20. The software recited in Claim 13 further comprising a code segment that programmatically enables and disables cacheability of a selected shadow RAM register.